Обучение машинного перевода без параллельных текстов*

Артеменков*[1] *А. А., Бахтеев*[1] *О. Ю., Стрижов*[2] *В. В.
[1]Московский физико-технический институт
[2]Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН

В данной работе исследуется задача машинного перевода между двумя языками. Предлагается подход, основанный на моделях автокодировщиков и не требующий наличия большого корпуса параллельных предложений. Каждому предложению из обоих языков ставится в соответствие вектор в общем скрытом пространстве. Оптимизация проводится таким образом, чтобы скрытые пространства автокодировщиков для разных языков совпадали. Для проверки качества модели проводится вычислительный эксперимент по переводу предложений между парой языков русский-украинский.

Ключевые слова: *нейронные сети, машинный перевод, автокодировщики.*

Введение

Целью данной работы является ьрешение задачи машинного перевода в отсутствии достаточного корпуса параллельных предложений. При наличии нескольких миллионов [?] параллельных образцов хорошо себя показывают методы машинного перевода с использованием нейронных сетей [?], [?]. Высокие результаты достигаются при использовании глубоких (свёрточных или рекуррентных) нейронных сетей, однако, в данном подходе критично наличие большой обучающей выборки. Частичное решение данной проблемы было найдено в пополнении числа предложений с помощью использования переводчиков более низкого качества. В [?] было показано, что данным способом могут быть улучшены результаты работы системы статистического машинного перевода Moses [?]. Более общим подходом является отказ от перевода в одну сторону и параллельное обучение переводчиков таким образом, чтобы один пополнял обучающую выборку другого.

Описанный выше метод был использована в [?] для перевода предложений с английского языка на французский. В данной работе подобная технология будет применяться для перевода с русского языка на украинский. Рассматриваются автокодировщики, реализованные в виде рекуррентных нейронных сетей [?], [?], используемые для прямого и обратного перевода, и сеть-дискриминатор, оптимизируемая с целью по представлению слова в векторном пространстве определять язык [?]. Автокодировщики оптимизируются таким образом, чтобы их латентные представления совпадали, или, что эквивалентно, чтобы дискриминатор не мог с достаточной уверенностью определить язык, соответствующий сгенерированному вектору. Для того, чтобы избежать переобучения, добавляется шум, не дающий автокодировщикам восстанавливать предложения в точности. Шаг оптимизации состоит из двух стадий: оптимизация дискриминатора и оптимизация переводчика. На первой стадии выбирается случайное предложение из исходного языка, кодируется с добавлением шума [?]) и подаётся на вход дискриминатору. После шага оптимизации аналогичные действия повторяются со случайным предложением из конечного языка. На второй стадии выбирается случайное предложение из исходного языка и переводится те-

---

Научный руководитель: Стрижов В. В. Авторы: А.В. Грабовой, О.Ю. Бахтеев, В.В. Стрижов, Eric Gaussier, координатор Малиновский Г. С. Консультант: Бахтеев О. Ю.

кущей версией переводчика на конечный язык. Затем на него накладывается шум, оно кодируется, и считываются показания дискриминатора. После шага оптимизации предложение переводится обратно в исходный язык и вычисляется значение функции потерь. После шага оптимизации действия повторяются со случайным предложением из конечного языка.

В качестве эксперимента производится перевод предложений с русского языка на украинский. Для этой пары языков отсутствует большие выборки параллельных предложений в открытом доступе, при этом достаточно данных по каждому из языков в отдельности. Качество полученного в результате переводчика оценивается с помощью метрики BLEU [?].

Постановка задачи

Рассматриваемая модель состоит из кодировщика f и декодировщика g, и отвечающих соответственно за отображение предложений из обоих языков в латентное пространство и обратное отображение из латентного пространства в предложения первого или второго языка. Кодировщик и декодировщик реализованы в виде рекуррентных нейронных сетей. Так как процедуры перевода из разных языков требуют разные словари, будем обозначать это индексами: f^{src} и g^{src} для автокодировщика первого языка и f^{tgt} и g^{tgt} для автокодировщика первого языка.

Обозначим через $\mathfrak{D}^{\text{src}} = [\mathbf{s}_1^{\text{src}}, \ldots, \mathbf{s}_{m_{\text{src}}}^{\text{src}}]$ корпус предложений из первого языка, через $\mathfrak{D}^{\text{tgt}} = [\mathbf{s}_1^{\text{tgt}}, \ldots, \mathbf{s}_{m_{\text{tgt}}}^{\text{tgt}}]$ – корпус предложений из второго языка, вообще говоря, не являющиеся параллельными. Также дана валидационная выборка $\mathfrak{D}^{valid} = \{(\mathbf{s}_1^{\text{src}}, \mathbf{s}_1^{\text{tgt}}), \ldots, (\mathbf{s}_{m_{\text{valid}}}^{\text{src}}, \mathbf{s}_{m_{\text{valid}}}^{\text{tgt}})\}$ представляющая собой корпус параллельных предложений. Также введём обозначения для мощностей словарей обоих языков: V^{src} и V^{tgt}. Тогда под $s(k)$ будем иметь в виду k-е слово предложения s, представленное в виде $x \in \mathbb{R}^{V^{\text{src}}}$ или $x \in \mathbb{R}^{V^{\text{tgt}}}$ соответственно. В качестве метрики между словами $d(\cdot, \cdot)$ предлагается использовать accuracy$(\cdot, \cdot)$ или cross-entropy$(\cdot, \cdot)$. Минимизируется следующая функция потерь:

$$L = \sum_{i=1}^{m_{\text{valid}}} \sum_{k=1}^{|\mathbf{s}_{m_{\text{valid}}}^{\text{tgt}}|} d(g^{\text{tgt}}(f^{\text{src}}(\mathbf{s}_{m_{\text{valid}}}^{\text{src}}))(k),\ \mathbf{s}_{m_{\text{valid}}}^{\text{tgt}}(k))$$

Так как во время обучения не используются пары параллельных предложений, то требуется добиться того, чтобы автокодировщики вели себя похожим образом. Этого можно добиться, потребовав сходство латентных пространств. Для этого предлагается использовать сеть-дискриминатор q, которая по латентному представлению h некоторого предложения определяет, какому языку оно принадлежит. Функция потерь записывается таким образом, чтобы оптимизировать дискриминатор r для распознавания представлений предложений, а автокодировщик – для генерации максимально похожих представлений слов из разных языков. Для того, чтобы избежать переобучения автокодировщика, каждый раз перед кодированием предложения к нему добавляется шум $\sigma(\cdot)$: из предложения опускаются некоторые слова, а к оставшимся применяется перестановка π таким образом, чтобы слова переставлялись не слишком далеко: $\max(\pi(i) - i) \leqslant k$. Окончательно функцию потерь можно записать следующим образом:

$$L_{AE} = d(g(f(\sigma(x))), x)$$
$$L_{TR} = d(g(f(\widehat{g}(f(x)))), x)$$
$$L_{ADV} = \log p(\text{язык} = src|Encoder(x)) + \log p(\text{язык} = tgt|Encoder(y))$$

(что бы последнее ни значило; уточнится после того, как будет написано какое-то количество кода)

$$L = aL_{AE} + bL_{TR} + cL_{ADV}$$